On Outage of Wireless Cloud Computing: Offloading Optimization and Bottleneck Analysis

Di Han*, Wei Chen*, Bo Bai†, Yuguang Fang‡
* Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Electronic Engineering, Tsinghua University, Beijing, CHINA, 100084
† Future Network Theory Lab, Huawei Technologies Co.,Ltd., Shatin, Hong Kong
‡ Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611,USA
Email: hd15@mails.tsinghua.edu.cn, wchen@tsinghua.edu.cn, baibo8@huawei.com, fang@ece.ufl.edu

***Abstract*—Wireless cloud computing system with computation offloading has attracted much attention due to the potential of alleviating the restrictions of limited resources in mobile devices. During the computation offloading, the steps of transmissions and computation need to be completed within the required time, otherwise the system will be in outage. Due to uncertainties of channel fading and computational complexities, such outage is the result of either transmission outage or computation outage. In this paper, a theoretical framework is proposed to analyze the outage of the wireless cloud computing system with multiple subcarriers and computation resources. Through theoretical analysis of the outage probability, the outage bottleneck can be identified. Numerical results have verified the theoretical analysis and concluded that the outage bottleneck depends not only on the distributions of complexities of tasks but also on the numbers of subcarriers and computation resources.**

I. Introduction

In recent years, the demand for mobile devices to run heavy applications with high computation requirement is increasing, while the resources of mobile devices, e.g., battery life and computation capability, are still extremely poor due to the limited physical size. Wireless cloud computing with computation offloading is considered as a potential technique to overcome these challenges [1][2] by transmitting insensitive and heavy computation tasks to resourceful computation resources in the cloud, which have attracted much attention from both academia and industries [3][4].

With the development of mobile networks and cloud computation, energy efficiency becomes an important design consideration due to the limited battery life of mobile devices. However, due to the uncertainties of wireless channels and computation requirements, the communication-computation tradeoff is much more challenging [5]. To better utilize the resource in computation offloading systems, there is a tradeoff in energy consumption between transmission and computation [6]. A framework for computation offloading based on execution time and energy consumption was proposed in [7], which aims to shorten response time and reduce energy consumption. In [8], the optimal timeout for local execution was found and an adaptive approach was proposed to save energy.

In this paper, we focus on a wireless cloud computing system which is composed of a mobile device connecting with multiple computation resources through multiple transmission subcarriers. In the process of offloading, not only the transmission of a task and its output results, but also the computation of the task in the cloud must be completed within the required time. Otherwise, timeout will be triggered, resulting in task resubmission, which we simply state that the system will be in outage state. Clearly, the outage event is composed of transmission outage and computation outage due to the uncertainties of channels (i.e., transmission subcarriers) and computation complexities of the task. To improve the system performance in terms of outage, both transmission and computation capabilities can be improved by increasing the power consumption of the system. However, in practice, the same outage performance may require different levels of power consumption in transmission and computation. Therefore, we consider the major factors that restricts the reduction of outage probability of the system even with the increase of power consumption, which will be referred to as the *outage bottleneck*. By analyzing the features of complexities of tasks and different offloading schemes, the necessary and sufficient conditions to determine the outage bottleneck of the system in two typical schemes are obtained, which can be used to evaluate the outage performance and select the most economical way to improve the systems performance for existing and upcoming wireless cloud computing system with computation offloading.

II. System Model and Problem Formulation

Consider a wireless cloud computing system with computation offloading that consists of one mobile device and M computation resources, whose set is denoted by $\mathcal{C} = \{c_1, \ldots, c_M\}$. The mobile device offloads the tasks to the computation resources in different locations over N $(N \geq M)$ orthogonal subcarriers, whose set is given by $\mathcal{S} = \{s_1, \ldots, s_N\}$.

A. Timeslot Model

As shown in Fig. 1, the time horizon is divided into timeslots, whose length is a fixed value T_{s}. Assume that each

This research was partially supported by the Chinese National 973 Program under Project 2013CB336600 and the National Natural Science Foundation of China under Project 61671269 and 61621091. The work of Y. Fang was partially supported by US National Science Foundation under grants CNS-1409797 and CNS-1343356.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

task of the mobile device arrives at the beginning of each timeslot, whose data size (e.g. in bytes) is a fixed value $l_{\rm in}$. Moreover, the data size of output result of each task is also a fixed value $l_{\rm out}$.

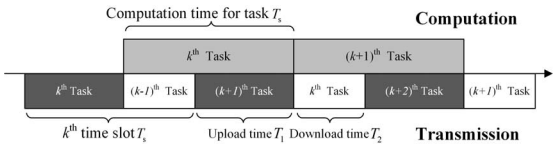

Fig. 1: The proposed upload and download structure.

Similar to our previous work [5], each timeslot can be divided into two parts: upload sub-timeslot T_1 and download sub-timeslot T_2. Clearly, $T_{\rm s} = T_1 + T_2$. As shown in Fig. 1, the mobile device is set to upload the k-th task to the cloud at the beginning of k-th timeslot and then download the corresponding output result from the cloud in the download sub-timeslot of the $(k+1)$-th timeslot. Therefore, the computation offloading process of each task can be divided into three steps: upload, computation and download, whose delay tolerances are T_1, T_2 and $T_{\rm s}$, respectively.

Since both of computation and transmission aspects play a key role in the computation offloading, we will introduce the computation and transmission models next. Without loss of generality, we focus on the offloading process and outage probability analysis of k-th task.

B. Computation Model

Let w denote the computation capability required by the whole task, which is a random variable with the probability density function $f_w(x)$ and could be measured by the required CPU cycles. Moreover, there also exists other background task in each computation resource. Let q_i denote the computation capability required by the background task in i-th computation resource during the computation time, which are independent with the probability density function $f_q(x)$. Moreover, w and q_i, $1 \le i \le M$, are independent with each other.

The computation capability of each computation resource is a finite value C, which could also be measured by the total number of CPU cycles in one timeslot T_s. Let $f_{\rm c}$ denote the clock frequency of the CPU in each computation resource. Moreover, according to [10][11], the power consumption of CPU in each computation resource can be expressed as

$$\xi_{\rm c} = \kappa {f_{\rm c}}^3, \tag{1}$$

where $\xi_{\rm c}$ is the power consumption of computation and κ is the effective switched capacitance depending on the chip architecture. Therefore, the computation capability C can be rewritten as

$$C(\xi_{\rm c}) = f_{\rm c} T_{\rm s} = k_c {\xi_{\rm c}}^{\frac{1}{3}}, \tag{2}$$

where $k_c = T_{\rm s}\kappa^{-\frac{1}{3}}$. Therefore, the percentage of the whole task can be completed by i-th computation resource during the computation step is given by

$$R_i^{\rm c}(w, q_i, \xi_{\rm c}) = \frac{C(\xi_{\rm c}) - q_i}{w}. \tag{3}$$

C. Transmission Model

Consider the block-fading channel, which means the channel state keeps fixed during a timeslot, and the channel states in different timeslots are independent and identically distributed. Since computation resources are located in different places, we assume the channels between the mobile device and different computation resources undergo independent fading. The channel gain matrix in the upload and download step are given by $\mathbf{H}^{\rm u} = [\mathbf{h}_1^{\rm u}, \ldots, \mathbf{h}_M^{\rm u}]$ and $\mathbf{H}^{\rm d} = [\mathbf{h}_1^{\rm d}, \ldots, \mathbf{h}_M^{\rm d}]$, where $\mathbf{h}_i^{\rm u} = [h_{i1}^{\rm u} \ldots, h_{iN}^{\rm u}]^T$ and $\mathbf{h}_i^{\rm d} = [h_{i1}^{\rm d} \ldots, h_{iN}^{\rm d}]^T$ denote the channel gain vector for i-th computation resource. Likewise, $h_{ij}^{\rm u}$ and $h_{ij}^{\rm u}$ are the channel gain of the upload and dowload link between the mobile device and i-th computation resource over j-th subcarrier, which are assumed to be independent of the identical distribution $\mathcal{CN}(0, 1)$.

In the upload step, to take full advantage of independent fading and maximize the diversity gain of transmission, the whole task will be broadcasted to all computation resources over N subcarriers. Therefore, the percentage of whole task can be uploaded to i-th computation resource is given by

$$R_i^{\rm u}(\mathbf{h}_i^{\rm u}, \xi_{\rm t}) = \frac{T_1 \log(1 + |\max\limits_j h_{ij}^{\rm u}|^2 \frac{\xi_{\rm t}}{N_0})}{l_{\rm in}}, \tag{4}$$

where $\xi_{\rm t}$ is the transmission power consumption, N_0 is the background noise power in each subcarrier.

In the download step, the subcarrier occupied by i-th computation resource in the download step is denoted $\mathbf{m}_i = [m_{i1}, \ldots, m_{iN}]^T$, where $m_{ij} = 1$ denotes that j-th subcarrier is allocated to i-th computation resource. Clearly, one subcarrier cannot be allocated to multiple computation resources in order to avoid collision, i.e., $\sum_{i=1}^{M} m_{ij} \le 1$, $j = 1, \ldots, N$. Similar to Eq. (4), the percentage of the result of input task can be downloaded to i-th computation resource after the download step is given by

$$R_i^{\rm d}(\mathbf{h}_i^{\rm d}, \mathbf{m_i}, \xi_{\rm t}) = \frac{T_2 \log(1 + |\max\limits_j \{m_j h_{ij}^{\rm d}\}|^2 \frac{\xi_{\rm t}}{N_0})}{l_{\rm out}}. \tag{5}$$

D. Outage of Wireless Cloud Computing

Assume that each task is splittable, which can be divided into multiple subtasks and then be processed in parallel on different computation resources. Then, the separate results of all subtasks can be combined into a whole result with negligible computation capability consumption [9], i.e., time consumption can be ignored. The corresponding percentages of the original task that is partitioned into the subtask on each computation resource are denoted by vector $\mathbf{x} = [x_1, \ldots, x_M]$. Clearly, to guarantee the quality of service, the sum of subtasks must cover the complete original task, i.e.,

$$\sum_{i=1}^{M} x_i = 1. \tag{6}$$

Similar to [9], we assume the original task is uniformly splittable, i.e., a fraction of x_i of the original task has $x_i l_{\rm in}$ input data bits, $x_i l_{\rm out}$ output data bits and requires $x_i w$ CPU

cycles for processing. Therefore, the percentage of the whole task that can be offloaded successfully is given by

$$\begin{aligned} &R(w,\mathbf{q},\mathbf{H}^{\mathrm{u}},\mathbf{H}^{\mathrm{d}},\mathbf{x},\mathbf{M},\xi_c,\xi_t)\\ =&\sum_{i=1}^{M}\min\{\mathbb{1}\left(R_i^{\mathrm{u}}(\mathbf{h}_i^{\mathrm{u}},\xi_t)\right),R_i^{\mathrm{d}}(\mathbf{h}_i^{\mathrm{d}},\mathbf{m_i},\xi_t),R_i^{\mathrm{c}}(w,q_i,\xi_{\mathrm{c}}),x_i\}, \end{aligned} \tag{7}$$

where $\mathbf{M}=[\mathbf{m}_1,\ldots,\mathbf{m}_M]$ and the indicator function $\mathbb{1}(a)$ is equal to one if $a<1$; otherwise, it is equal to zero. Thus, the system will be in outage state if

$$R(w,\mathbf{q},\mathbf{H}^{\mathrm{u}},\mathbf{H}^{\mathrm{d}},\mathbf{x},\mathbf{M},\xi_c,\xi_t)<1, \tag{8}$$

which means this task cannot be completed fully within the required time. Moreover, $\mathbf{x}$ is a design parameter, which can be determined based on $\mathbf{q}$ and $\mathbf{H}^{\mathrm{u}}$ after upload step.

III. Outage Probability Analysis

In this section, we will focus on the outage probability of wireless cloud computing system with given M and N in different offloding schemes, i.e., different vector $\mathbf{x}$.

By recalling Eqs. (4)(7-8), the probability of the whole task uploaded to each computation resource successfully in upload step is given by

$$p_{\mathrm{b}}^{\mathrm{u}}(\xi_{\mathrm{t}})=\Pr\left\{R_i^{\mathrm{u}}(\mathbf{h}_i^{\mathrm{u}},\xi_{\mathrm{t}})<1\right\}. \tag{9}$$

Therefore, the probability of the whole task received by m $(0\le m\le M)$ computation resources successfully is given by

$$p_{\mathrm{b}}^{\mathrm{u}}(m,\xi_{\mathrm{t}})=\left(\begin{array}{c}M\\m\end{array}\right)\left(1-p_{\mathrm{b}}^{\mathrm{u}}(\xi_{\mathrm{t}})\right)^m p_{\mathrm{b}}^{\mathrm{u}}\left(\xi_{\mathrm{t}}\right)^{(M-m)}. \tag{10}$$

Clearly, if $m=0$, i.e., no computation resource have received the whole task, the system is considered in outage.

Since each task can be processed in parallel on multiple computation resources, we consider two typical offloading schemes, i.e., Cloud scheme and Fog scheme, which are summarized in Table I.

TABLE I: Two Computation Offloading Schemes

	Upload	Computation	Download
Cloud scheme	Broadcast	Centralized	Point-to-point
Fog scheme	Broadcast	Distributed	Multiple access

Consider the scenario that the whole task have been received by m $(1\le m\le M)$ computation resources after upload step whose set is denoted by $\mathcal{C}_{\mathrm{ava}}=\{d_1,\ldots,d_m\}$, where $\mathcal{C}_{\mathrm{ava}}\in\mathcal{C}$ and $|\mathcal{C}_{\mathrm{ava}}|=m$. Then, the two offloading schemes can be defined as follows:

1) Cloud scheme: Select one computation resource with maximal available computation capability, i.e., minimal background task, whose subscript is i^* to process the whole task. In this scheme, we have $x_i=1$ if and only if $i=i^*$, and otherwise $x_i=0$. Thus, the topology of this scheme is similar to that of cloud computing.
2) Fog scheme: Divide the whole task into m identical portions, i.e., subtasks, to each available computation resource in $\mathcal{C}_{\mathrm{ava}}$ and be processed in parallel. In this scheme, $x_i=\frac{1}{m}$ if and only if $c_i\in\mathcal{C}_{\mathrm{ava}}$, and otherwise $x_i=0$. Thus, the topology of this scheme is similar to that of fog computing.

Then, we analyze the outage probability of two considered typical schemes, respectively.

A. Cloud Scheme with Centralized Computation

By recalling Eqs. (2-3)(7-8), the probability of the whole task that cannot be completed within required time in Cloud scheme is given by

$$\begin{aligned} p_{\mathrm{cen}}^{\mathrm{c}}(m,\xi_{\mathrm{c}})&=\Pr\left\{\max_{c_i\in\mathcal{C}_{\mathrm{ava}}}R_i^{\mathrm{c}}(w,q_i,\xi_{\mathrm{c}})<x_i\right\}\\ &=\Pr\left\{w+\min_{i=1,\ldots,m}q_i>C(\xi_{\mathrm{c}})\right\}. \end{aligned} \tag{11}$$

In download step, the result of the whole task will be transmitted from the i^*-th computation resource to the mobile device, which is point-to-point transmission over N subcarriers, i.e., $\mathbf{m}_{i^*}=[1,\ldots,1]$. Since $x_{i^*}=1$, the probability of result cannot be downloaded with required time is given by

$$p_{\mathrm{cen}}^{\mathrm{d}}(\xi_{\mathrm{t}})=\Pr\left\{R_{i^*}^{\mathrm{d}}(\mathbf{h}_{i^*}^{\mathrm{d}},\mathbf{m}_{\mathbf{i}^*},\xi_{\mathrm{t}})<1\right\}. \tag{12}$$

When $p_{\mathrm{cen}}^{\mathrm{c}}(m,\xi_{\mathrm{c}})$ and $p_{\mathrm{cen}}^{\mathrm{d}}(N,\xi_{\mathrm{t}})$ are small enough, i.e., ξ_{c} and ξ_{t} are large, the probability of the computation and dowload steps cannot be completed with m available computation resources in Cloud scheme can be approximated by

$$p_{\mathrm{cloud}}(m,\xi_{\mathrm{t}},\xi_{\mathrm{c}})\approx p_{\mathrm{cen}}^{\mathrm{c}}(m,\xi_{\mathrm{c}})+p_{\mathrm{cen}}^{\mathrm{d}}(\xi_{\mathrm{t}}), \tag{13}$$

Therefore, the outage probability of Cloud Scheme can be approximated by

$$\begin{aligned} &p_{\mathrm{cloud}}(\xi_{\mathrm{t}},\xi_{\mathrm{c}})\\ \approx&p_{\mathrm{b}}^{\mathrm{u}}(0,\xi_{\mathrm{t}})+\sum_{m=1}^{M}p_{\mathrm{b}}^{\mathrm{u}}(m,\xi_{\mathrm{t}})\left(p_{\mathrm{cen}}^{\mathrm{c}}(m,\xi_{\mathrm{c}})+p_{\mathrm{cen}}^{\mathrm{d}}(\xi_{\mathrm{t}})\right). \end{aligned} \tag{14}$$

Then, by recalling Eqs. (11-14), the following theorem can be obtained.

Theorem 1. *When ξ_{c} and ξ_{t} are large, we have*

$$p_{\mathrm{cloud}}(\xi_{\mathrm{t}},\xi_{\mathrm{c}})\approx k_1{\xi_{\mathrm{t}}}^{-N}+p_{\mathrm{cen}}^{\mathrm{c}}(M,\xi_{\mathrm{c}}), \tag{15}$$

where k_1 is a constant and

$${N_0}^N 2^{NR_2}\le k_1\le {N_0}^N\left(2^{NR_1}M+2^{NR_2}\right), \tag{16}$$

where $R_1=\frac{l_{\mathrm{in}}}{T_1}$ and $R_2=\frac{l_{\mathrm{out}}}{T_2}$.

Proof. When ξ_{c} and ξ_{t} are large, by recalling Eqs. (9)(12), we have

$$\begin{cases} p_{\mathrm{b}}^{\mathrm{u}}(\xi_{\mathrm{t}})=\left(1-e^{-2^{R_1}\frac{N_0}{\xi_{\mathrm{t}}}}\right)^N\approx\left(2^{R_1}\dfrac{N_0}{\xi_{\mathrm{t}}}\right)^N;\\ p_{\mathrm{cen}}^{\mathrm{d}}(\xi_{\mathrm{t}})=\left(1-e^{-2^{R_2}\frac{N_0}{\xi_{\mathrm{t}}}}\right)^N\approx\left(2^{R_2}\dfrac{N_0}{\xi_{\mathrm{t}}}\right)^N \end{cases} \tag{17}$$

Moreover, we have

$$p_{\mathrm{cen}}^{\mathrm{c}}(m,\xi_{\mathrm{c}})\in[0,1],\forall m=1,\ldots,M. \tag{18}$$

The theorem is proven by consisting Eqs. (17-18) in Eq. (19). □

$$
\begin{aligned}
p_{\text{cloud}}(\xi_{\text{t}},\xi_{\text{c}}) &\approx \left(2^{R_1}\frac{N_0}{\xi_{\text{t}}}\right)^{NL} + \sum_{m=1}^{M}\binom{M}{m}\left(1-\left(2^{R_1}\frac{N_0}{\xi_{\text{t}}}\right)^{N}\right)^{m}\left(2^{R_1}\frac{N_0}{\xi_{\text{t}}}\right)^{N(M-m)}\left(p^{\text{c}}_{\text{cen}}(m,\xi_{\text{t}})+\left(2^{R_2}\frac{N_0}{\xi_{\text{t}}}\right)^{N}\right)\\
&\approx \sum_{m=1}^{M}\binom{M}{m}\left(2^{R_1}\frac{N_0}{\xi_{\text{t}}}\right)^{N(M-m)}\left(p^{\text{c}}_{\text{cen}}(m,\xi_{\text{t}})+\left(2^{R_2}\frac{N_0}{\xi_{\text{t}}}\right)^{N}\right)\\
&\approx p^{\text{c}}_{\text{cen}}(M,\xi_{\text{t}})+\left(2^{R_2}\frac{N_0}{\xi_{\text{t}}}\right)^{N}+M\left(2^{R_1}\frac{N_0}{\xi_{\text{t}}}\right)^{N}p^{\text{c}}_{\text{cen}}(M-1,\xi_{\text{t}})\\
&\in \left(p^{\text{c}}_{\text{cen}}(M,\xi_{\text{t}})+\left({N_0}^{N}2^{NR_2}\right)\xi_{\text{t}}^{-N}, p^{\text{c}}_{\text{cen}}(M,\xi_{\text{t}})+{N_0}^{N}\left(2^{NR_1}M+2^{NR_2}\right)\xi_{\text{t}}^{-N}\right).
\end{aligned}
\tag{19}
$$

B. Fog Scheme with Uniformly Distributed Computation

By recalling Eqs. (2-3)(7-8), the probability of not all subtasks can be processed within required time is given by

$$
\begin{aligned}
p^{\text{c}}_{\text{dis}}(m,\xi_{\text{c}}) &= \Pr\left\{\min_{c_i\in\mathcal{C}_{\text{ava}}} R^{\text{c}}_i(w,q_i,\xi_{\text{c}}) < x_i\right\}\\
&= \Pr\left(\frac{w}{m}+\max_{i=1,\ldots,m} q_i > C(\xi_{\text{c}})\right).
\end{aligned}
\tag{20}
$$

After computation step, the result of each subtask will be transmitted from the corresponding computation resource in $\mathcal{C}_{\text{ava}}$ to the mobile device in download step, which is multiple access transmission of m computation resources over N subcarriers ($m \le M \le N$). Since $x_i = \frac{1}{m}$ for $c_i \in \mathcal{C}_{\text{ava}}$, the probability of not all results of the subtasks downloaded within required time is given by

$$
p^{\text{d}}_{\text{dis}}(m,\xi_{\text{t}}) = \Pr\left\{\min_{c_i\in\mathcal{C}_{\text{ava}}} R^{\text{d}}_i(\mathbf{h}^{\text{d}}_i,\mathbf{m}_i,\xi_{\text{t}}) < \frac{1}{m}\right\}. \tag{21}
$$

The optimal $\mathbf{m}_i$, $1 \le i \le M$ of minimizing $p^{\text{d}}_{\text{dis}}(m,\xi_{\text{t}})$ can be obtained by the $\mathbf{R^2HK}$ algorithm in [12]. Moreover, according to the Lemma 6 and Theorem 1 in [12], after optimal subcarrier allocation based on the $\mathbf{R^2HK}$ algorithm, $p^{\text{d}}_{\text{dis}}(m,\xi_{\text{t}})$ in high ξ_{t} regime, i.e., ξ_{t} are large, is given by

$$
p^{\text{d}}_{\text{dis}}(m,\xi_{\text{t}}) = a_0(m)m\left(p^{\text{s}}_{\text{dis}}(m,\xi_{\text{t}})\right)^{N} + o\left(p^{\text{s}}_{\text{dis}}(m,\xi_{\text{t}})\right), \tag{22}
$$

where $o(x)$ denotes the higher order infinitesimal of x,

$$
a_0(m) = \begin{cases} 1, & m < N\\ 2, & m = N, \end{cases} \tag{23}
$$

and

$$
p^{\text{s}}_{\text{dis}}(m,\xi_{\text{t}}) = \Pr\left\{R^{\text{d}}_i(\mathbf{h}^{\text{d}}_i,[1,0,\ldots,0],\xi_{\text{t}}) < \frac{1}{m}\right\}, \tag{24}
$$

Similar to Cloud scheme, when ξ_{c} and ξ_{t} are large, the outage probability of Fog scheme can be approximated by

$$
\begin{aligned}
&p_{\text{fog}}(\xi_{\text{t}},\xi_{\text{c}})\\
\approx& p^{\text{u}}_{\text{b}}(0,\xi_{\text{t}}) + \sum_{m=1}^{M} p^{\text{u}}_{\text{b}}(m,\xi_{\text{t}})\left(p^{\text{c}}_{\text{dis}}(m,\xi_{\text{c}})+p^{\text{d}}_{\text{dis}}(m,\xi_{\text{t}})\right).
\end{aligned}
\tag{25}
$$

Similar to Theorem 1, we also have the following theorem in Fog scheme.

Theorem 2. *When ξ_{c} and ξ_{t} are large, we have*

$$
p_{\text{fog}}(\xi_{\text{t}},\xi_{\text{c}}) \approx k_2\xi_{\text{t}}^{-N} + p^{\text{c}}_{\text{dis}}(M,\xi_{\text{c}}), \tag{26}
$$

where k_2 is a constant and

$$
a_0(M)M{N_0}^{N}2^{NR^{\text{d}}_2} \le k_2 \le a_0(M)M{N_0}^{N}\left(2^{NR_1}+2^{NR^{\text{d}}_2}\right). \tag{27}
$$

IV. Outage Bottleneck Analysis

According to Theorem 1 and Theorem 2, both of the outage probabilities of two schemes with high power consumption of computation and transmission can be divided into two parts, which are related to transmission and computation capacities, respectively. Therefore, we have the following definitions.

Definition 1. The computation and transmission outage probability partitions of Cloud and Fog scheme are defined by

$$
\begin{cases} p^{\text{c}}_1(M,\xi_c) = p^{\text{c}}_{\text{cen}}(M,\xi_{\text{c}});\\ p^{\text{t}}_1(N,\xi_t) = \xi_{\text{t}}^{-N}. \end{cases} \tag{28}
$$

and

$$
\begin{cases} p^{\text{c}}_2(M,\xi_c) = p^{\text{c}}_{\text{dis}}(M,\xi_{\text{c}});\\ p^{\text{t}}_2(N,\xi_t) = \xi_{\text{t}}^{-N}. \end{cases} \tag{29}
$$

Then, for each given M and N, the power consumptions with $p^{\text{c}}_k(M,\xi_c) = p^{\text{t}}_k(N,\xi_t) = \varepsilon$ can be written as

$$
\begin{cases} \xi_{\text{c}}(\varepsilon) = {p^{\text{c}}_k}^{-1}(\varepsilon) = f_k(\varepsilon);\\ \xi_{\text{t}}(\varepsilon) = {p^{\text{t}}_k}^{-1}(\varepsilon) = g_k(\varepsilon), \end{cases} \tag{30}
$$

where $p^{-1}(x)$ is the inverse function of $p(x)$ and $k \in \{1,2\}$. In this work, the major part of total power consumption as $\varepsilon \to 0$ is considered as the *outage bottleneck*, which is the major factor of restricting the decline of outage probability of the system by increasing total power consumption. Then, by recalling Eq. (30), the outage bottleneck of this system can be found by comparing power consumption ratio

$$
h_k(\varepsilon) = \frac{\xi_{\text{c}}(\varepsilon)}{\xi_{\text{t}}(\varepsilon)} = \frac{f_k(\varepsilon)}{g_k(\varepsilon)} \tag{31}
$$

with $\varepsilon \to 0$, where $k \in \{1,2\}$.

Definition 2. Define[1]

$$\Xi_k = \lim_{\varepsilon \to 0} h_k(\varepsilon) = \lim_{\varepsilon \to 0} \frac{f_k(\varepsilon)}{g_k(\varepsilon)} = \lim_{\xi_c \to \infty} \frac{\xi_c}{g_k({f_k}^{-1}(\xi_c))}, \quad (32)$$

where $k \in \{1, 2\}$. If $\Xi_k = 0$ or ∞, the outage bottleneck of corresponding scheme is only transmission or computation capability, respectively. Otherwise, if $\Xi_k \in (0, \infty)$, the outage bottleneck are both transmission and computation capability.

The rest of this paper is focused on the analysis of the features of Ξ_k. Then, we have the following theorem.

Theorem 3.

$$\Xi_k = \lim_{x \to \infty} \frac{1}{{k_c}^3} \left\{ \frac{f_{u_k}(x)}{3Nx^{-(3N+1)}} \right\}^{\frac{1}{N}}, \quad (33)$$

where $f_{u_k}(t)$ denotes the probability density function of u_k, which is given by

$$u_k = \begin{cases} w + \min\limits_{i=1,\dots,M} q_i, & k = 1; \\ \dfrac{w}{M} + \max\limits_{i=1,\dots,M} q_i, & k = 2. \end{cases} \quad (34)$$

Proof. By recalling Eqs. (11)(20)(28-30), we have

$$g_k({f_k}^{-1}(\xi_c)) = \frac{1}{\Pr\{u_k > k_c x^{\frac{1}{3}}\}^{\frac{1}{N}}}. \quad (35)$$

By recalling Eq. (32) and using L'Hôpital's rule, we have

$$\begin{aligned} \Xi_k &= \lim_{x \to \infty} \frac{x}{g_k({f_k}^{-1}(x))} \\ &= \lim_{x \to \infty} \frac{1}{{k_c}^3} \left\{ \frac{\Pr\{u_k > x\}}{x^{-3N}} \right\}^{\frac{1}{N}} \\ &= \lim_{x \to \infty} \frac{1}{{k_c}^3} \left\{ \frac{f_{u_k}(x)}{3Nx^{-(3N+1)}} \right\}^{\frac{1}{N}}, \end{aligned} \quad (36)$$

□

According to Theorem 3, the outage bottleneck is depend on the distribution of u_k, $k \in \{1, 2\}$. Then, by analyzing the relationship between the distributions of w, q_i and u_k, the outage bottleneck can be obtained effectively.

Definition 3. Define

$$\begin{cases} \varphi_1 = \lim\limits_{x \to \infty} x^{3N+1} f_w(x); \\ \phi_1 = \lim\limits_{x \to \infty} x^{\frac{3N}{M}+1} f_q(x), \end{cases} \quad (37)$$

where $f_w(x)$ and $f_q(x)$ are the probability density functions of w and q_i, respectively.

Theorem 4. *If $\max\{\varphi_1, \phi_1\} = 0$ or ∞, the bottleneck in Cloud scheme is only transmission or computation capability, respectively. Otherwise, if $\max\{\varphi_1, \phi_1\} \in (0, \infty)$, the outage bottleneck is both transmission and computation capability.*

[1]In this paper, the scenario that the limitation $\lim\limits_{\varepsilon \to 0} h_k(\varepsilon)$ does no exist is not considered.

Proof. According to the definition in Eq. (34), since w and q_i are non-negative random variables and independent with each other, we have

$$\begin{cases} \Pr\{u_1 > x\} \geq \Pr\{w > x\} \text{ or } \Pr\{q > x\}; \\ \Pr\{u_1 > x\} \leq \Pr\left\{w > \frac{x}{2}\right\} + \Pr\left\{q > \frac{x}{2}\right\}, \end{cases} \quad (38)$$

where $q = \min\limits_{i=1,\dots,M} q_i$. By substituting Eq. (38) into Ξ_1 in Eq. (36), we have

$$\begin{cases} \Xi_1 \geq \lim\limits_{x \to \infty} \left\{ \frac{\Pr\left\{w \geq k_c x^{\frac{1}{3}}\right\}}{x^{-N}} \right\}^{\frac{1}{N}}; \\ \Xi_1 \geq \lim\limits_{x \to \infty} \left\{ \frac{\Pr\left\{q \geq k_c x^{\frac{1}{3}}\right\}}{x^{-N}} \right\}^{\frac{1}{N}} = \lim\limits_{x \to \infty} \left\{ \frac{\Pr\left\{q_i \geq k_c x^{\frac{1}{3}}\right\}}{x^{-\frac{N}{M}}} \right\}^{\frac{M}{N}}; \\ \Xi_1 \leq \lim\limits_{x \to \infty} \left\{ \frac{\Pr\left\{w \geq \frac{k_c x^{\frac{1}{3}}}{2^{\frac{1}{3}}}\right\} + \Pr\left\{q \geq \frac{k_c x^{\frac{1}{3}}}{2^{\frac{1}{3}}}\right\}}{x^{-N}} \right\}^{\frac{1}{N}}. \end{cases} \quad (39)$$

Then, by using L'Hôpital's rule, the inequalities in Eq. (39) can be rewritten as

$$\begin{cases} \Xi_1 \geq \lim\limits_{x \to \infty} \frac{1}{{k_c}^3} \left\{ \frac{f_w(x)}{3Nx^{-(3N+1)}} \right\}^{\frac{1}{N}} = k_c^{-3} k_1 {\varphi_1}^{\frac{1}{N}}; \\ \Xi_1 \geq \lim\limits_{x \to \infty} \frac{1}{{k_c}^3} \left\{ \frac{f_q(x)}{\frac{3N}{M} x^{-(\frac{3N}{M}+1)}} \right\}^{\frac{M}{N}} = k_c^{-3} k_2 {\phi_1}^{\frac{1}{N}}; \\ \Xi_1 \leq 8 k_c^{-3} \left({k_1}^N \varphi_1 + {k_2}^N \phi_1 \right)^{\frac{1}{N}}, \end{cases} \quad (40)$$

where $k_1 = (3N)^{-\frac{1}{N}}$ and $k_2 = \left(\frac{3N}{M}\right)^{-\frac{M}{N}}$. Therefore, the Ξ_1 can be bounded by

$$k_c^{-3} \theta^{\frac{1}{N}} \leq \Xi_1 \leq 8 k_c^{-3} (2\theta)^{\frac{1}{N}}, \quad (41)$$

where $\theta = \max\{{k_1}^N \varphi_1, {k_2}^N \phi_1\}$. This theorem is proven. □

Similar to Definition 3 and Theorem 4 in Cloud scheme, we have the following definition for Fog scheme.

Definition 4. Define

$$\begin{cases} \varphi_2 = \lim\limits_{x \to \infty} x^{3N+1} f_w(x) = \varphi_1; \\ \phi_2 = \lim\limits_{x \to \infty} x^{3N+1} f_q(x). \end{cases} \quad (42)$$

Theorem 5. *If $\max\{\varphi_2, \phi_2\} = 0$ or ∞, the bottleneck in Fog Scheme is only transmission or computation capability, respectively. Otherwise, if $\max\{\varphi_2, \phi_2\} \in (0, \infty)$, the outage bottleneck is both transmission and computation capability.*

V. Numerical Results

Without loss of generality, we set $k_c = 1$ with two typical cases of the distributions of w and q_i , which are given by

1) Case 1: w and q_i follow the identical independent Gamma distribution with $\alpha = \beta = 1$, i.e., exponential distribution with parameter $\lambda = 1$.

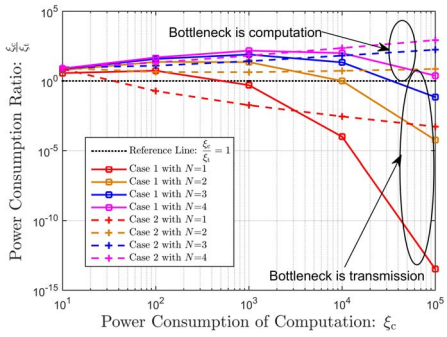

Fig. 2: Power consumption ratio with $M = 1$.

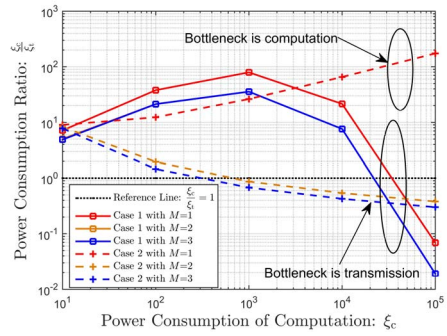

Fig. 3: Power consumption ratio with $N = 3$ in Cloud scheme.

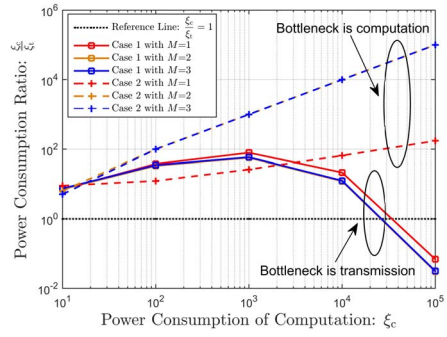

Fig. 4: Power consumption ratio with $N = 3$ in Fog scheme.

2) Case 2: w and q_i follow the independent power-law distribution, whose probability density functions are $f_w(x) = \frac{10}{x^{11}}$ and $f_q(x) = \frac{5}{x^6}$ for $x \geq 1$, respectively.

Firstly, we consider the scenario that $M = 1$, i.e., Fog scheme degrades into Cloud scheme since there is only one computation resource in the cloud. Fig. 2 presents the power consumption ratio $\frac{\xi_c}{\xi_t}$ defined in Eq. (31) with the assurance that $p_k^t(\xi_t) = p_k^c(\xi_c)$, $k \in \{1, 2\}$, with the increase of ξ_c. By increasing the number of subcarriers N from 1 to 4, the power consumption ratio of case 1 always tends to 0, while that of case 2 tends to ∞ when $N \geq 2$ with the increase of ξ_c. It shows that the outage bottleneck for case 1 with $1 \leq N \leq 4$ and case 2 with $N = 1$ is transmission capability and for case 2 with $N \geq 2$ is computation capability, which verifies the theoretical analysis and results in Theorem 4 and Theorem 5.

Then we consider the scenario that $N = 3$. Figs. 3 and 4 present the power consumption ratio $\frac{\xi_c}{\xi_t}$ with the assurance that $p_k^t(\xi_t) = p_k^c(\xi_c)$, $k \in \{1, 2\}$, with the increase of ξ_c in Cloud scheme and Fog scheme, respectively. As shown in Fig. 3, by increasing the number of computation resources M from 1 to 3, the power consumption ratio of case 1 always tends to 0, while that of case 2 with $M \leq 1$ tends to ∞ with the increase of ξ_c in Cloud scheme. However, Fig. 4 shows that the power consumption ratio of case 1 and case 2 with $1 \leq M \leq 3$ in Fog scheme always tends to 0 and ∞, respectively. Thus, the numerical results in Figs. 3 and 4 also verify Theorem 4 and Theorem 5.

VI. Conclusion

In this paper, we have developed a theoretical framework for analyzing the outage bottleneck of a wireless cloud computing system, which is the major factor that limits the reduction of outage probability even with the increase of the total power consumption. By analyzing the distributions of channel gains and computation complexities and bounding the outage probability, the detailed decision conditions to identify the outage bottleneck in two typical offloading schemes, i.e., centralized computation and distributed computation, are obtained. The theoretical results can be used to evaluate the outage performance and bottleneck for existing and upcoming wireless cloud computing systems and improve system performance via the most economical way.

References

[1] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51-56, Apr. 2010.

[2] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones," *Proc. MobiCASE*, Oct. 2010.

[3] K. Kumar, J. Liu, Y. Lu, and B. Bhargave,"A survey of computation offloading for mobile systems," *Mobile. Netw. Appl.,* vol. 18, no. 1, pp. 129-140, Feb. 2013.

[4] C. Wang and Z. Li, "A computation offloading scheme on handheld devices," *J. Parallel. Distrib. Comput.,* vol. 64, no. 6, pp. 740-746, Jun. 2004.

[5] D. Han, B. Bai, and W. Chen, "Outage bottleneck for reliable mobile computation offloading transmission or computation?" *Proc. IEEE GlobalSIP*, Dec. 2016.

[6] X. Ma, Y. Cui, L. Wang, and I. Stojmenovic, "Energy optimizations for mobile terminals via computation offloading", *Proc. PDGC*, Dec. 2012.

[7] Y. D. Lin, E. H. Chu, Y. C. Lai, and T. J. Huang, "Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds," *IEEE Syst. J.,* vol, 9, no. 2, pp. 393-405, Jun. 2015.

[8] C. Xian, Y. H. Lu, and Z. Y. Li, "Adaptive computation offloading for energy conservation on battery-powered systems, *Proc. ICPAD*, Dec. 2007.

[9] G. Calice, A. Mtibaa, R. Beraldi, and H. Alnuweiri, "Mobile-to-mobile opportunistic task splitting and offloading," *Proc. WiMob*, 2015.

[10] T. Burd and R. Broderson, "Processor design for portable systems," *J. VLSI Singapore Process*, vol. 13, nos. 2-3, pp. 203-222, Aug. 1996.

[11] S. Kaxiras and M. Martonosi, *Computer Architecture Techniques for Power-Efficiency.* Morgan and Claypool, 2008.

[12] B. Bai, W. Chen, Z. Cao, and K. Letaief, "Max-matching diversity in OFDMA systems", *IEEE Trans. Commum.*, vol. 58, no. 4, pp.1161-1171, Apr. 2010.

[13] J. R. Lorch and A. J. Smith, "Improving dynamic voltage scaling algorithms with pace," *Proc. ACM SIGMETRICS*, pp. 50-61.

[14] S. Foss, D. Korshunov, and S. Zachary, *An introduction to heavy-tailed and subexponential Distributions*, New York: Springer, 2011.